科学研究費助成事業　　研究成果報告書

平成　２６　年　　６　月　１０　日現在

機関番号：　１２６０８
研究種目：　基盤研究(C)
研究期間：　2011～2013
課題番号：　２３５０００３４

研究課題名（和文）汎言語的メタプログラミング基盤としての健全な構文マクロ機構の研究

研究課題名（英文）A study on hygienic syntactic macro system as a generic meta-programming foundation

研究代表者
脇田　建（Wakita, Ken）

東京工業大学・情報理工学（系）研究科・准教授

研究者番号：１０２４２２６５

交付決定額（研究期間全体）：（直接経費）　　3,900,000円、（間接経費）　　1,170,000円

研究成果の概要（和文）：マクロシステムはさまざまなプログラミング言語で使用されているが，多くの問題の原因となることも指摘されている．この点を改善するHygienic構文マクロシステムはLISPについて研究されてきたが，一般のプログラミング言語への応用は限定的であるため，われわれはこの系統的な実装方式にうちて研究し，その技術を応用してJavaScriput，およびScalaのためのHygienic構文マクロシステムを完成させた．研究成果として拡張可能な構文解析器の実装，および，汎用マクロ展開器の実装という二つの主要な困難を解決した．研究成果としてWebで実装を公開している．

研究成果の概要（英文）：Macro systems have been widely adopted to existing programming languages. However, it is also known that simple macro systems cause serious programming errors. Hygienic syntactic macro system that has been studied in the context of LISP culture is a solution to problems of traditional macro system but a implementations for non-LISP languages are rare and a systematic implementation scheme has been unknown. The research proposes a compact and systematic implementation scheme for generic programming languages. We have applied the proposed technique to implement hygienic macro systems for JavaScript and Scala. The proposal resolves two technical obstacles: (1) implementation of extensible parser, and (2) implementation of macro expander.

研究分野：　総合領域

科研費の分科・細目：　情報学・ソフトウェア

キーワード：　構文マクロシステム　メタプログラミング　JavaScript　LISP

１．研究開始当初の背景

本研究が目指すのは，プログラマが目的に応じてプログラミング言語の構文を自由に拡張できる仕組みを汎用マクロシステムとして提供することである．素朴なマクロシステムは 60 年代に提案されたが，本研究が対象とする構文マクロは 70-80 年代に Lisp 文化で培われたより高度な構文マクロである．構文マクロは構文木の変換システムであり，非常に複雑な構文の定義が可能となる．一例をあげると Common Lisp のオブジェトシステム(CLOS)の構文は構文マクロシステムで記述されている．80 年代後半に提案された変数参照に関する健全性(Hygienic 性)の効率的な実装方法が 90 年代に見つかって一応の完成を見た．

健全な構文マクロの有用性の利便性は認められたものの，それを Lisp ではない言語に取り込むための研究は Dylan を除けば存在しない．

２．研究の目的

本研究の目標は，プログラミング言語を特定せずに健全な構文マクロシステムを効率的に実装するための一般的な手法を見つけることとともにそのマクロシステムの表現力をドメイン特化型言語のマクロ実装を通して示す点にある．本研究の特徴は以下の通りである．

汎言語性　文法，意味論に依存せずプログラミング言語一般に適用可能

平易　パターン照合を用いた一般のプログラマにも容易なマクロシステム

実装方式　Scheme との相互変換にもとづくコンパクトな実装

健全性　構文構造と変数参照の意味を保持する健全性(Syntactic & Hygienic)

LISP において構文マクロをはじめとするメタプログラミングが花開いた技術的な素地は「プログラムはデータ」にある。これと引き換えに LISP は多くの人が嫌悪する括弧だらけの S 式を採用した．一方，他の普通のプログラミング言語におけるメタプログラミングでは、言語の構文と大幅に異なる抽象構文木を扱うことが求められる．このため、メタプログラミングはコンパイラ開発者のような一部の専門家の専売特許となっている．

LISP 以外の言語に構文マクロを導入する上での課題は以下の二つである．

拡張可能な構文解析器　構文マクロ機構はプログラミング言語に対する構文の拡張機能である．マクロを用いて新たに導入される構文に対応するためには，プログラミング言語の構文解析器はそれらの構文を正しく構文解析できなくてはならない．

汎用マクロ展開器　Scheme のマクロ展開器は約 30,000 行の非常に複雑なコードである．これを別のプログラミング言語で利用するための簡易かつ系統的な実装方式が求められる．

３．研究の方法

平成 23 年度は提案の心臓部にあたる JavaScript 向けのマクロ展開器の実装に取り組んだ．研究の準備段階から温めていた着想を元にプロトタイプを実装し，JavaScript の式，分，宣言に関わるマクロ定義を扱えるようになった．また，この応用として役割指向プログラミングという新しい手法への適用可能性を探り，その方向性から社会シミュレーション用言語 SOARS の実装を試みることで，ドメイン特化型言語の実装についてのマクロ機構の有効性を検証した．

平成 24 年度は拡張可能な構文解析器の実装方式として，解析表現文法を抽象層として取り入れることによって，ソフトウェアアーキテクチャを明瞭化できることを示

した．また，この方式で実装を修正することによって既存研究に比べてはるかに表現力に優れたマクロシステムをコンパクトに実装できることを示した．また，マクロ展開器については，それを直接実装するのではなく Scheme のマクロ展開器に移譲するための相互プログラム変換の技術を確立した．

平成 25 年度は研究をさらに深化させ，C++のような伝統的なオブジェクト指向言語，C のような言語仕様に多くの制約がある言語，Scala のような関数型とオブジェクト指向が融合したパラダイムについての Hygienic マクロシステムの応用について検討を重ね，それぞれの問題点と対応方法について検討した．さらに，Scala についてはこの考察を踏まえてマクロシステムを完成することができた．

４．研究成果

マクロシステムはさまざまなプログラミング言語で使用されているが，多くの問題の原因となることも指摘されている．この点を改善する Hygienic 構文マクロシステムは LISP について研究されてきたが，一般のプログラミング言語への応用は限定的であるため，われわれはこの系統的な実装方式について研究し，その技術を応用して JavaScript，および Scala のための Hygienic 構文マクロシステムを完成させた．研究成果として拡張可能な構文解析器の実装，および，汎用マクロ展開器の実装という二つの主要な困難を解決した．研究成果として実装を公開している．

以下は本研究のプロトタイプシステムとして Hygienic 構文マクロシステムを拡張した JavaScript プログラミングにおけるマクロ定義の例である．

```
expression 旅程 {
  symbol: A, B, X;
  symbol: から，まで，で;
  { 旅程 [# A から B まで X で #], ... =>
    [{ from: A, to: B, by: X }, ...] }
```

ここにおいて「旅程」と命名された JavaScript の式として利用可能なマクロが定義されている．ExJS のマクロ定義では，そのなかで使用されるメタ変数とキーワードが宣言される．ここでは，A, B, X がメタ変数に該当し，それぞれ任意の JavaScript の式を表す．また，キーワードとしては「から」,「まで」,「で」を利用できる．宣言に続いて，マクロ形式が定義されている．ここで，旅程のあとに「A から B まで X で」という字句の繰り返しであることが宣言されている．

さらに，このマクロ形式がどのように展開されてどのような JavaScript のコードが出力されるかが記述されている．ここでは，一連の旅程を表す字句が JavaScript のオブジェクトの配列に変換されることが記述されている．

上に定義されたマクロを利用したマクロの使用例は以下のとおりである．

```
var 私の旅程 = 旅程
    大岡山 から 目黒駅 まで 目黒線 で
    目黒駅 から 東京駅 まで 山手線 で
    東京駅 から 京都駅 まで 新幹線 で;
```

このように一見すると日本語かと思われるような記述が許されるようになったが，これを変換した JavaScript は以下のようなものとなる．

```
var 私の旅程 = [ { from: “大岡山”, to: “目黒駅”, by: “大井町線” }, { from: “目黒駅”, to: “東京駅”, by: “山手線” }, { from: “東京駅”, to: “京都駅”, by: “新幹線” } ];
```

すでに述べたようにマクロの定義から，拡張構文の解析器が生成され，その拡張構文で解析された結果は一旦 JSON で表現されたのち，S 式に変換され，そこでのマ

クロ展開を経てマクロを含まないプログラムのＳ式表現が得られる．このＳ式で表されたプログラムを JavaScript に変換することによって，マクロ展開が完了する．

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計　1 件）

(1) 甫水佳奈子，脇田 建，佐々木 晃，解析表現文法と Scheme マクロ展開器を用いた JavaScript 向け Hygienic 構文マクロシステムの実装，情報処理学会論文誌．プログラミング 6(2)，85-101，2013，査読有．
http://id.nii.ac.jp/1001/00094924/

〔学会発表〕（計　14 件）

(1) 栗原 あずさ，佐々木 晃，脇田 建，ビジュアルブロックを採用したドメイン特化言語とその開発ツールの実現手法，電子情報通信学会ソフトウェアサイエンス研究会，2014 年 3 月 12 日，那覇，査読なし．
http://www.ieice.org/ken/paper/20140312aBML/

(2) 森 健輔，脇田 建，差分的に記述された解析表現文法に対する構文解析器の合成，日本ソフトウェア科学会創設 30 周年記念大会，2013 年 9 月 13 日，東京，査読なし．
http://www.is.titech.ac.jp/~wakita/sites/jssst2013/program.pdf

(3) 柏木 孝仁，佐々木 晃，田沼 英樹，役割指向を用いた並行システムの記述と実装手法の提案，第 3 回社会システム部会研究会，2013 年 2 月 27 日，盛岡，査読なし．
http://www.socsys.org/symposium003/program/

(4) 脇田 建，甫水 佳奈子，佐々木晃，Hygienic 構文マクロシステムを用いた JavaScript プログラミング，第 54 回プログラミング・シンポジウム，2013 年 1 月 11 日，箱根，査読なし．
http://www.ipsj.or.jp/prosym/54/54program.html

(5) Kanako Homizu, Ken Wakita, and Akira Sasaki, A proposal of implementation technique for hygienic syntactic macro system for JavaScript, 10th Asian symposium on Programming languages and systems, 2012 年 12 月 12 日，京都，査読なし．

(6) 柏木 孝仁，佐々木 晃，田沼 英樹，Scheme 言語をルール記述言語とした役割指向記述の試み，情報処理学会全国大会，2012 年 3 月 6 日，名古屋，査読なし．
http://id.nii.ac.jp/1001/00084585/

(7) 甫水 佳奈子，脇田 建，佐々木 晃，構文マクロ定義を利用した動的拡張可能なエディタ，日本ソフトウェア科学会第 19 回インタラクティブシステムとソフトウェアに関するワークショップ，2011 年 12 月 12 日，宮津，査読なし．
http://www.wiss.org/WISS2011Proceedings/PDF/066.pdf

〔その他〕
ホームページ等

- 甫水 佳奈子，js-macro: A hygienic macro system for JavaScript
  https://github.com/homizu/js-macro
- 甫水 佳奈子，マクロプログラミングサンプル集
  http://www.is.titech.ac.jp/~homizu8/ex-js/index.html
- 脇田 建，高級マクロシステム
  http://kwakita.wordpress.com/projects/macros

６．研究組織
(1)研究代表者
研究代表者：脇田 建（Wakita, Ken）
東京工業大学・情報理工学研究科・准教授
研究者番号：10242265

(2)研究分担者
研究分担者：佐々木 晃（Sasaki, Akira）
法政大学・情報科学部・准教授
研究者番号：90396870